Q1) WAP to retrieve and display all the employees working in a given department entered by the user at run from an existing table employee in a database sample

```
import mysql.connector as sql


connection = sql.connect(host="127.0.0.1", user="root", database="emp", password="xxx")
c = connection.cursor()

c.execute(f"SELECT * FROM employee WHERE deptno={input('Enter Department Number: ')}")
data = [e[1] for e in c.fetchall()]
print("\n".join(data if data else ["No record found"]))
```

OUTPUT

Enter Department Number: 20
Smith
Jones
Scott
Adams
Ford

Enter Department Number: 1
No record found

Q2) WAP to create a table game (game no, name , nop) and insert rows in it using user defined functions create_table() and insert_rows()

```
import mysql.connector as sql

connection = sql.connect(host="127.0.0.1", user="root", password="Sujal@5243", database="assignment",
autocommit=True)
c = connection.cursor()

def create_table(table_name, schema):
    try:
        c.execute(f"CREATE TABLE {table_name} ({','.join(schema)});")
    except Exception as e:
        print("An error occurred:\n", e)

def insert_values(table_name, values):
    try:
        c.execute(f"INSERT INTO {table_name} VALUES({','.join(values)});")
    except Exception as e:
        print("An error occurred:\n", e)

create_table("game", (
    "game_no int primary key",
    "game_name varchar(20)",
    "nop int"
))

while True:
    insert_values("game", (
        input("Enter game no: "),
        f'"{input("Enter game name: ")}"',
        input("Enter no. of players: ")
    ))
    if input("Continue (y/n): ").lower() ≠ "y":
        break

connection.close()
```

OUTPUT

Enter game no: 1
Enter game name: football
Enter no. of players: 22
Continue (y/n): y
Enter game no: 2
Enter game name: basketball
Enter no. of players: 18
Continue (y/n): y
Enter game no: 3
Enter game name: volleyball
Enter no. of players: 12
Continue (y/n): n

Q3) WAP to interface between python and mysql to do the following

1) to create a table game in mysql
2) add records in by taking values at runtime from user
3) to display all students who have got a particular grade which is input by the user

```
import mysql.connector as sql


connection = sql.connect(host="127.0.0.1", user="root", password="Sujal@5243", database="assignment",
autocommit=True)
c = connection.cursor()


def create_table(table_name, schema):
    try:
        c.execute(f"CREATE TABLE {table_name} ({','.join(schema)});")
    except Exception as e:
        print("An error occurred:\n", e)


def insert_values(table_name, values):
    try:
        c.execute(f"INSERT INTO {table_name} VALUES({','.join(values)});")
    except Exception as e:
        print("An error occurred:\n", e)


create_table("game", (
    "class int",
    "name varchar(20) not null",
    "game varchar(20)",
    "grade varchar(20) not null"
))


while True:
    insert_values("game", (
        input("Enter class number: "),
        f'"{input("Enter student name: ")}"',
        f'"{input("Enter game: ")}"',
        f'"{input("Enter grade: ").upper()}"'
    ))
    if input("Continue adding? (y/n): ").lower() ≠ "y":
        break


while True:
    grade = input("Enter grade to look for: ")
    c.execute(f"SELECT name FROM game WHERE grade='{grade.upper()}';")
    table = [std[0] for std in c.fetchall()]
    print("\n".join(table if table else ["No record found"]))
    if input("Continue searching? (y/n): ").lower() ≠ "y":
        break


connection.close()
```

OUTPUT

Enter class number: 10
Enter student name: sujal
Enter game: football
Enter grade: a
Continue adding? (y/n): y
Enter class number: 11

Enter student name: feynman
Enter game: basketball
Enter grade: a
Continue adding? (y/n): y
Enter class number: 12
Enter student name: hilbert
Enter game: volleyball
Enter grade: b
Continue adding? (y/n): n
Enter grade to look for: a
sujal
feynman
Continue searching? (y/n): n

Q4) WAP to:
a) create the given table orders
b) add values to the table with values from user taken at runtime
c) delete a record from table for a given order number
d) display all records

```
import mysql.connector as sql


connection = sql.connect(host="127.0.0.1", user="root", password="Sujal@5243", database="assignment",
autocommit=True)
c = connection.cursor()


def create_table(table_name, schema):
    try:
        c.execute(f"CREATE TABLE {table_name} ({','.join(schema)});")
    except Exception as e:
        print("An error occurred:\n", e)


def insert_values(table_name, values):
    try:
        c.execute(f"INSERT INTO {table_name} VALUES({','.join(values)});")
    except Exception as e:
        print("An error occurred:\n", e)


def delete_order_by_no(order_no, table_name="orders"):
    try:
        c.execute(f"DELETE FROM {table_name} WHERE order_no={order_no}")
    except Exception as e:
        print("An error occurred:\n", e)


def display_record(table_name):
    try:
        c.execute(f"SELECT * FROM {table_name};")
        data = [c.column_names] + c.fetchall()
        space, buffer = [], []

        # GET PADDING
        for column in range(len(data[0])):
            for row in range(len(data)):
                buffer.append(len(str(data[row][column])))
            space, buffer = space + [max(buffer)], []

        table_width = sum([extra + 4 for extra in space])

        # PRINT OUT TABLE
        print("━" * table_width)

        for row in range(len(data)):
            print("┃ ", end="")
            for column in range(len(data[row])):
                entry = str(data[row][column])
                entry = entry + " " * (space[column] - len(entry))
                print(f"{entry} ┃  ", end="")
            print("\n", "━" * table_width, sep="")

        # print("‾" * table_width)
    except Exception as e:
        print("An error occurred:\n", e)


create_table("orders", (
```

```
    "order_no int(10)",
    "client_name varchar(30)",
    "client_loc varchar(30)",
    "orders int(10)",
    "payments int(10)"
))

while True:
    insert_values("orders", (
        input("Enter Order no: "),
        f'"{input("Enter client name: ")}"',
        f'"{input("Enter client loc: ")}"',
        input("Enter orders: "),
        input("Enter payments: ")
    ))
    if input("Continue adding? (y/n): ").lower() ≠ "y":
        break

while True:
    delete_order_by_no(int(input("Enter order no to delete: ")))
    if input("Continue deleting? (y/n): ").lower() ≠ "y":
        break

display_record("orders")
connection.close()
```

OUTPUT

Enter Order no: 1
Enter client name: sujal
Enter client loc: test
Enter orders: 2
Enter payments: 3
Continue adding? (y/n): y

Enter Order no: 2
Enter client name: feynman
Enter client loc: test
Enter orders: 3
Enter payments: 4
Continue adding? (y/n): y

Enter Order no: 3
Enter client name: hilbert
Enter client loc: test
Enter orders: 4
Enter payments: 5
Continue adding? (y/n): y

Enter Order no: 4
Enter client name: neumann
Enter client loc: test
Enter orders: 5
Enter payments: 6
Continue adding? (y/n): n

Enter order no to delete: 3
Continue deleting? (y/n): n

order_no	client_name	client_loc	orders	payments
1	sujal	test	2	3
2	feynman	test	3	4
4	neumann	test	5	6